Learning Robot Control using a Hierarchical SOM-based Encoding

Georgios Pierris, *NCSR Demokritos,* and Torbjørn S. Dahl, *Plymouth University,*

***Abstract*—Hierarchical representations and modeling of sensorimotor observations is a fundamental approach for the development of scalable robot control strategies. Previously, we introduced the novel Hierarchical Self-Organizing Map-based Encoding algorithm (HSOME) that is based on a computational model of infant cognition. Each layer is a temporally augmented SOM and every node updates a decaying activation value. The bottom level encodes sensori-motor instances while their temporal associations are hierarchically built on the layers above. In the past, HSOME has shown to support hierarchical encoding of sequential sensor-actuator observations both in abstract domains and real humanoid robots. Two novel features are presented here starting with the novel skill acquisition in the complex domain of learning a double tap tactile gesture between two humanoid robots. During reproduction, the robot can either perform a double tap or prioritize to receive a higher reward by performing a single tap instead. Secondly, HSOME has been extended to recall past observations and reproduce rhythmic patterns in the absence of input relevant to the joints by priming initially the reproduction of specific skills with an input. We also demonstrate in simulation how a complex behavior emerges from the automatic reuse of distinct oscillatory swimming demonstrations of a robotic salamander.**

***Index Terms*—Robot Programming by Demonstration, Artificial Neural Networks, Self-Organizing Maps, Tactile Gestures**

I. Introduction

Robot *Programming by Demonstration* (PbD) studies the problem of encoding and representing complex motor skills into compact mathematical formulations. PbD accelerates the process of learning motor skills; first, by decreasing the time and expertise needed in order to provide a high quality demonstration to a robot, and secondly, by reducing the problem to fitting models on the already acquired demonstrations, rather than exploring the full space. A necessary feature for the development of the next generation of autonomous robots is the capability of these algorithms to autonomously invent novel skills and behaviors that derive from the demonstrated ones. In this work, we present such an extension to a previously published work that enables a robot a) to learn a specific skill from demonstration, b) the development of a novel skill that is derived from the on-line modulation of a known skill during action selection, and c) the emergence of a complex behavior from the autonomous reuse of distinct skills. Consequently, this work touches upon distinct directions that would otherwise call for different algorithms. However, being inspired by the learning processes of infants, we present a unified learning architecture to address these needs.

Different machine learning paradigms have been proposed in the literature, such as supervised, unsupervised, and reinforcement learning, to facilitate skill learning in robots. However, robots need a combination of paradigms to bootstrap learning and then to self-improve the demonstrated skill [1], or to better generalize a skill in order to adapt in small variations [2]. The study of infant cognitive development remains a promising research area to draw inspiration from, and the influence of Piaget's constructivism theory has been instrumental to robotics [3]. A common ground in computational models of constructivism is their reliance on hierarchical representations of knowledge or skills. The hierarchical representation of skills in particular, allows agents to derive complex models of novel skills by efficiently reusing already acquired skills. In turn, these new complex skills may be reused again on a higher level to derive even more complex skills. This process of acquiring complex skills is extensively studied in the area of Developmental Robotics that tries to bridge the gap between the traditional developmental areas of psychology and neuroscience with robotics [4], [5]. In return, robots are transformed to research platforms that cognitive scientists apply, test, and validate embodied models of cognition and development [6]. However, there are other directions in infant cognition that need to be studied if we want to better understand the learning processes of human brains. In particular, there is evidence, based on similarities between Central Pattern Generators and microcircuits that the evolution of locomotion control has influenced the development of higher cognition [7].

Cohen *et al.* [8] introduced a computational model of infant cognition, i.e., the Constructivist Learning Architecture (CLA), an Information Processing approach suggesting that infant knowledge is the result of processing information in the environment. In previous work, we introduced a novel algorithm based on the CLA architecture that facilitated motor skill learning [9], [10]. The proposed *Hierarchical Self-Organizing Map-based Encoding* (HSOME) algorithm is structurally built as a hierarchy of temporally augmented layers of Kohonen *Self-Organizing Maps* (SOM). The latter use node-specific decaying activation values that represent the short-term memory. The weights of the nodes represent the long-term memory.

Dr Georgios Pierris is with the Institute of Informatics and Telecommunications at the National Centre for Scientific Research, "Demokritos", Athens, 15310, Greece, e-mail: gpierris@iit.demokritos.gr. The research leading to these results was performed while being affiliated with the University of South Wales, UK.

Dr Torbjørn S. Dahl is with the Centre for Robotics and Neural Systems, at Plymouth University, Plymouth, United Kingdom, e-mail: torbjorn.dahl@plymouth.ac.uk

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013, Challenge 2 - Cognitive Systems, Interaction, Robotics, under grant agreement 231500-ROBOSKIN, and has been partially funded by EUCogIII.

Manuscript received January 17, 2017.

By hierarchically connecting the sparse activations of a SOM as input to another layer the hierarchy follows the unitary-store memory model suggesting that the short-term memory consists of brief activations of long-term memory representations. Combined they build an auto-associative memory that encodes and recalls arbitrary observed sequences. HSOME supports sequential action selection, hidden state identification, learning of complex physical humanoid robot control, and novel skill acquisition by exploiting the demonstrated skills through Reinforcement Learning. Concretely, it was demonstrated that the HSOME algorithm (technical details are presented in Section III) can support learning in abstract domains (where robot observations and actions are represented by integers) and physical humanoid robots learning tactile gestures. HSOME successfully resolved ambiguity problems in hidden states during decision making and seamlessly recovered from perturbations internal (e.g., frictional forces on the skin and in the joints) and external (i.e., a human or artificial push) to the system [9], [10]. While a large part of the aforementioned research is focusing on humanoid robots and high level cognition, there is another research direction showing a great interest in biologically inspired robotics to approach the problem of learning motor skills from an evolutionary perspective. In particular, the evolutionary transition from aquatic environments to terrestrial locomotion is an interesting area to study that may answer questions with regards to the development of biological organisms [11]. Inspired by the salamander gaits that are produced by a *Central Pattern Generator* (CPG), a salamander-like robot has been developed, both physical and simulated versions, to study the development of CPGs for locomotion gaits production both for aquatic and terrestrial environments [11], [12].

In this work, we bridge these two directions into a single learning framework and present novel capabilities of the HSOME algorithm and further discuss how HSOME comes to support these features. In particular, we present HSOME in the context of a) learning motor skills in a physical humanoid robot for tasks that include a hidden state and null velocities, b) developing novel motor skills during action selection in a physical humanoid robot that have not been demonstrated before by a user using reinforcement learning, and c) the emergence of complex behavior in a simulated robot by selectively reusing a cyclic movement pattern from a repertoire of encoded skills. In order to build autonomous agents we must bridge the gap between the various research directions into developing learning frameworks that a) simultaneously support sequential decision making for discrete and rhythmic patterns, b) feature an auto-associative memory model to resolve temporal order and duration of events, and c) include goal-orientation in their behavior to further improve or develop novel skills. The proposed experimental setups, even when studied in isolation, represent classes of problems that the state-of-the-art algorithms have yet to conquer. However, what is more interesting in this work is the development of a single learning framework to support all three experimental setups.

The remainder of this paper is structured as follows. Section II presents a brief introduction to the state-of-the-art algorithms in learning motor control, the necessary literature that our proposed algorithm builds upon, and the literature in the area of rhythmic movement generation. Section III discusses the Hierarchical SOM-based Encoding algorithm (HSOME) and presents a novel mode of operation that enables the hierarchy to encode and reproduce oscillatory movements, establishing the ground for complex behaviors to emerge without explicitly being programmed. Section IV discusses in detail the experimental setup and procedure that was followed in the tactile gesture learning experiment and the simulated robotic salamander. The results and a discussion on each experiment is presented in Section V. Finally, Section VI concludes this work and proposes extensions that could improve the results in both experiments.

II. Background

The structure of the literature review initially describes the broader area of learning robot control. The review continues with the connectionist approaches and continues with the literature most related to this work and more specifically in hierarchical Self-Organizing Maps. Finally, as we present an experiment on rhythmic movement learning and generation, an additional section is devoted to the area of learning rhythmic motions and more specifically to the CPGs that are being used for imitation learning in our experimental setups.

A. Learning Robot Control

Learning by demonstration (or Imitation Learning) is an intuitive process of learning and reproducing motion patterns both for humans and robots. Imitation Learning enables an agent to take advantage of the high quality demonstrated actions in order to fullfil a given task. For humans the process feels trivial. In robotics, the process initially requires the encoding of a set of trajectories in a model and, later, the retrieval of the generalized motions. One such method is the *Gaussian Mixture Models* (GMM) [2], [13] that encode motion patterns into a compact representation of only a few Gaussians. Robots are then able to reproduce the desired motor skills through *Gaussian Mixture Regression* (GMR). However, several limitations are well-known in the literature for GMMs, e.g., learning multiple motor skills of different length or demonstrations that are not temporally and spatially aligned [14], as Gaussians cannot optimally fit the patterns due to high variance of the trajectories. Another method, the *Dynamic Movement Primitives* (DMPs) [15] offer a simplified representation of complex movements into a number of nonlinear dynamical systems, hence, DMPs need only a few parameters to learn a movement compared to encoding the raw high dimensional temporal signals. DMPs have shown to learn discrete movements, e.g., reaching a point or a tennis forehand swing and repeated movement patterns, e.g., drumming behavior. However, the aim of this work is to answer the question of how is it possible for a robot to move beyond the standard encoding of a motor skill, to either self-discovering novel motor skills based on the demonstrated ones or even developing new ones.

The research area of *Reinforcement Learning* (RL) represents a family of algorithms where agents attempt to learn an

optimal behavior in a dynamic environment through trial-and-error. RL offers a variety of algorithms to solve efficiently various tasks in abstract environments, however, robot control remains a challenging field for RL. An example of such a challenge is the inherently continuous space of states and actions of robotic environments. In the literature, various algorithms have been proposed to tackle this problem, e.g., with applications in learning an optimal controller for an under-actuated swinging-up pendulum and the cart-pole swing-up [16]. However, robots also exhibit high dimensionality that requires exponentially more data to cover the complete state-action space. At the same time, performing trials with physical robots is considerably more expensive [17]. Consequently, DMPs have dominated the RL literature as a policy representation framework that allows for imitation learning, smooth movement reproduction, and simplified representation of complex movement with only a few parameters compared to using the raw high dimensional temporal signals. The idea behind a DMP is to model basic motor primitives within a stable dynamical system, whose analytic properties are well-understood, e.g., a damped spring model, and then modulate it to accommodate the desired motions, e.g., the demonstrated tasks. Some of the most successful robotic applications of RL are the Ball-in-a-Cup game with a Barrett WAM™ [18], [17], the iCub learning the skill of archery [19], the pancake flipping skill with a Barrett WAM™ [20], and the simulated 50 DOFs planar robot and the simulated 12 DOFs robot dog jump [21]. What is common in these methods is the use of a simple underlying representation of the policy, such as the DMPs; which raises the question to what extent the RL algorithms are responsible for these interesting results or whether the oversimplification of representing the movements in a DMP is their advantage [22]. *Inverse Reinforcement Learning* (IRL) provides a framework for learning complex behaviors from expert demonstrations [23] in problems where it may be difficult or impossible to write an explicit reward function. IRL has been successfully employed in parking lot and urban navigation, human goal inference and other low dimensional state-action spaces, however, we are not aware of any literature in learning fine control in physical humanoid robots. Finally, two recent surveys specifically studying the domain of robotics are available in the literature both for Imitation Learning [24], and for Reinforcement Learning [17] approaches.

B. Connectionist Approaches

Recurrent Neural Networks (RNN) belong to a family of non-linear approximators that model dynamical systems through recurrent connections between neurons. RNNs handle efficiently temporal signals and approximate any non-linear dynamical system with arbitrary precision [25]. RNNs have been successful in various applications such as text generation [26], navigational tasks [27], and humanoid robot control [28]. However, a common drawback of RNNs is their computationally expensive training procedures. The *Reservoir Computing* (RC) paradigm alleviates the training problem of RNNs by training only a linear read-out of a randomly initialized dynamical system, i.e., the *reservoir*. *Echo State Networks* (ESNs) [29] fall into the RC paradigm and have been used in robot control to learn stationary object grasping and also to predict human walking motions [25]. However, experimental studies suggest that the prediction capabilities exhibited by ESNs in noiseless chaotic time-series prediction cannot be reproduced for noisy real time-series [30].

C. Related Work

1) Self-Orgazining Maps: A Kohonen *Self-Organizing Map* (SOM) [31] is an unsupervised algorithm that quantizes high dimensional data in a latent space of lower dimension preserving data topology. The technical details of SOMs are presented in Section III-A. Even though SOMs cannot encode temporal information, various extensions have been proposed in the literature [32], [33], [34]. However, the length of sequences that can be learned and the complexity of each update of the map's weights are both in the order of the size of the map. In order to address this limitation, hierarchical versions of temporal SOMs have been proposed, where the activation of past winning neurons is used as input to another map.

Inspired by findings on infant cognition, the *Constructivist Learning Architecture* (CLA) uses a layered SOM architecture with a persistent short-term memory activation model [8], in order to learn topographic maps of recent activations. CLA was demonstrated both to an abstract ball collision domain as well as in a mobile robot learning simulated environment [35]. A three-level Kohonen SOM has also been used in mobile robots to learn novel behavior plans [36], and also to recognize human gestures [37]. HSOME [9], [10] benefits from such a hierarchical representation increasing its power of representation exponentially with the number of layers. HSOME is a more general approach than its predecessor in that its structure is generic with no formal limitations on the height of the hierarchy and with no hand coded connections. On the application side, HSOME has been able to encode and reproduce motion patterns in physical humanoid robots that perform arbitrary movements and task specific tactile gestures in a robot-robot interaction scenario that have not been shown by other algorithms based on hierarchical SOMs. However, this work goes beyond previous applications of hierarchical SOMs in that we use HSOME for: 1) autonomous tactile gestures reproduction that feature null velocities in the hidden states, 2) autonomous novel skill acquisition in a humanoid robot, 3) autonomous emergence of complex swimming behavior with obstacle avoidance capabilities through the continuous reproduction of learned rhythmic movements. Technical details of the HSOME algorithm are discussed in Section III.

2) Rhythmic Patterns: Our understanding of how invertebrate and vertebrate animals produce rhythmic movements has become clearer in the last 30 years [38], [39]. Exploring and understanding the underlying principles of voluntary and involuntary movement in animals and humans is an important milestone towards building cognitive robots [11]. A special category of interest is the production of voluntary rhythmic motor patterns, e.g., walking, swimming, or chewing.

It is widely accepted that such movements across vertebrates are produced by autonomous neural networks that can

endogenously produce high-dimensional rhythmic patterns in the absence of rhythmic input; these are called Central Pattern Generators. For example, there is evidence that an isolated spinal cord from the body of the primitive fish *lamprey* can produce *fictive locomotion*; i.e., motor movements without explicit need of feedback, only using, e.g, chemical stimulation [40]. The existence of CPGs is also reported in other animals, e.g., young frog tadpoles [41] and salamanders [40]. CPGs produce patterns of activity in the absence of rhythmic input, but what is more interesting is how these patterns may be modulated by the sensory input. Hence, organisms are able to shape high dimensional patterns through low-dimensional input signals, e.g., a salamander can turn while swimming by shaping its motor pattern through simple signals.

Ijspeert *et al.* [11] developed both a real and a simulated robotic model of a salamander. Based on the biological evidence, a CPG model is distributed along the spinal cord of the salamander-like robot. The CPG is implemented as a collection of non-linear oscillators with controlled amplitude whose outputs are the joint angles for each joint across the spine. A 6DOF spine is implemented with 4 extra oscillators for the limbs that help the salamander walk in terrestrial environments. The complete CPG for the whole robot is controlled by two signals, the left and right drive. This controller is used in the swimming salamander experiment as a method to collect high quality training data before demonstrating the emergence of complex swimming behavior in the simulated salamander.

III. Hierarchical SOM-based Encoding

Various attempts have been made to resolve the limitation of traditional Kohonen SOMs to process sequential data [42]. The proposed Hierarchical SOM-based Encoding algorithm uses decaying neuron activations of past winning neurons in a hierarchy of SOMs. A node on every layer learns a compact representation of decayed activations from the level below. The resulting architecture is built on top of a biologically-inspired computational model of infant cognition [9], [10].

A. Background

A Kohonen SOM is an unsupervised data quantization tool that represents high dimensional data in a latent space of lower dimension. The interesting feature of a SOM is that it tries to preserve the data topology between the two spaces. Hence, nearby points in the input space are mapped to neighboring points in the latent space. However, this is not always possible [43].

In general, a SOM is a collection of nodes distributed in a lattice. Each node j holds a vector of trainable weights $\boldsymbol{w}_j$ of length equal to the input space. The nodes compete against each other based on a similarity measure, e.g., the inverse Euclidean distance, to respond to an input $\boldsymbol{x}$, therefore, SOMs belong to the family of *unsupervised competitive learning networks*.

The updated synaptic weights of neuron j at the next time step for a randomly drawn input $\boldsymbol{x_i}$ from the training set is defined by

$$\boldsymbol{w}_j(t+1) = \boldsymbol{w}_j(t) + a(t)h_{j,bmu(\boldsymbol{x_i})}(t)(\boldsymbol{x_i} - \boldsymbol{w}_j(t)). \quad (1)$$

The learning rate is decreasing over time and is defined as,

$$a(t) = a(t=0)\exp\left(-\frac{t}{\tau'}\right), \quad (2)$$

where τ' is a time constant that regulates the profile of the decrease. The update rule applies to all neurons with the maximum effect on the winning neuron and less effect as the lateral distance increases according to the neighborhood function, i.e., $h_{j,bmu(\boldsymbol{x})}$, where $bmu(\boldsymbol{x})$ is the index of the winning neuron for input $\boldsymbol{x}$.

B. From a Single Layer to a Hierarchy

The fundamental unit in HSOME is a temporally augmented SOM, i.e., a traditional SOM where each neuron i maintains on every step, t, a decaying activation value,

$$\alpha_{i,t} = A\alpha_{i,t-1} + I, \quad (3)$$

where $I = 2^{D-1}$ for the winning node, $I = 0$ for the remaining nodes and $A = 1/2$ (integer division). D is the *Short-term Memory* (STM) capacity after which an activated node is forgotten. The node-specific activation values follow the leaky integrator activation function [44]. The capacity of the STM is limited, hence, the introduction of another level above increases the encoding capabilities of the algorithm. An instance of an activation map can now be encoded in the level above after D timesteps. Nodes on the level above also get activated producing another activation map on that level, that will be updated every D^2 timesteps as well. An overview of the presented architecture is illustrated in Figure 1.

C. Training Algorithm

The training algorithm (Figure 2) is divided into two steps, namely the bottom layer training, and the training of the sequence-encoding layers above.

1) Bottom Layer: In its general case, robot control is formalized as the problem of performing an optimal action at a given observation, while receiving a reward in return. The same formalization is applied in HSOME. Assume $\hat{\boldsymbol{\xi}} = \{\xi_i(t)\}_{i=1}^{\kappa}$ be the set of κ demonstrations. Each i demonstration, $\xi_i(t)$, is a temporally ordered sequence of observations of length T_i, $\xi_i(t) = \{\xi(t)\}_{i=0}^{T_i}$. Each sample $\xi(t')$ represents a triplet of <observation, action, reward> for timestep t'. For specific setups where the action corresponds to actual joint configurations, it might be possible to omit the action; however, we are interested in the general case for the definition of the algorithm. The bottom level SOM is trained following the traditional SOM learning algorithm (line 9 in Figure 2). Randomly selected samples are drawn separately, with each one encoding the aforementioned triplet. As a result, the bottom level SOM corresponds to a discretized version of the initial hyperspace with each node representing a discrete observation, action and reward triplet.

2) Sequence-Encoding Layers: The layers above the bottom layer are the sequence-encoding layers and their purpose is to encode sequences of activations from the level below. Every D^l steps (line 12, Figure 2), where $l = 0, 1, 2\ldots$ is the height of the layer with zero being the bottom one, one node is

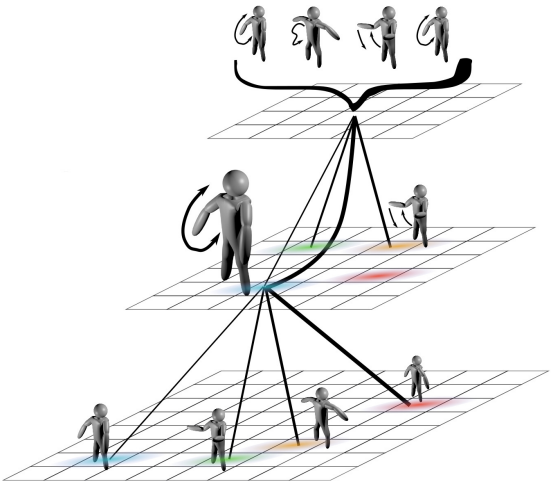


Fig. 1: The bottom layer augmented SOM provides a discretized representation of the high dimensional input space into instances of robot observation, action, and reward. Each sample of the ordered demonstrations activate a winning neuron (red) that decays on the following steps (orange $\rightarrow$ green $\rightarrow$ blue) unless the same node gets activated. The activations correspond to the short term memory of the SOM. The winning neuron on any layer higher than the bottom one, encodes on its weights a sparse activation map from the layer below. In the middle layer, the blue node represents the node reuse feature under different contexts.

1: $\hat{\boldsymbol{\xi}}$, κ kinesthetic demonstrations
2: D, STM capacity
3: L, total number of levels
4: $n^{(l)}$, number of nodes on level l
5: $\mathbf{M}_j^{(l)}$, activation of node j on level l
6: $\mathbf{w}_j^{(l)}$, weights of node j at level l
7: $\boldsymbol{o}_t$, observation at time step t
8: **procedure** TRAINING
9: TRAINBOTTOMSOM($\hat{\boldsymbol{\xi}}$) //Bottom layer training
10: **for each** $\xi_i(t)$ **do** //Sequence Encoding
11: ACTIVATENODE($\underset{j}{\operatorname{argmin}}\|\hat{\xi}_i(t) - \mathbf{w}_j^{(0)}\|$, $l = 0$)
12: **if** $t \mod D^l = 0$ **then** // $\forall l \in \{1, \ldots, L-1\}$
13: **if** $\min_j \|\mathbf{M}^{(l-1)} - \mathbf{w}_j^{(l)}\| = 0$ **then**
14: $k \leftarrow \underset{j}{\operatorname{argmin}}\|\mathbf{M}^{(l-1)} - \mathbf{w}_j^{(l)}\|$
15: **else**
16: $k \leftarrow$RETURNUNOCCUPIEDNODE(l)
17: $\mathbf{w}_k^{(l)} \leftarrow \mathbf{M}^{(l-1)}$
18: ACTIVATENODE(k, l)
19: **procedure** ACTIVATENODE(j, l)
20: $\forall i \neq j, M_i^{(l)} \leftarrow M_i^{(l)}/2$
21: $M_j^{(l)} \leftarrow I + M_j^{(l)}/2$

Fig. 2: Training Algorithm

selected as a winner at level l and updates its weights to encode an instance of the activation map below at level $l-1$. The winning node also gets activated (line 18, Figure 2) to form another activation pattern in that level. Hence, the training samples of each demonstration are processed sequentially to encode the temporal information in the higher levels. Node reuse is also possible if two common activation maps (line 13, Figure 2) have been formed on a layer at different stages of the demonstrations. Any node reuse is reflected to the activation value and it is possible to retrieve it at a later stage.

D. On-line Reproduction

In the post-training state of the hierarchy, there are initially no activations on any level. Each node at the bottom layer encodes an instance of the state-action-reward triplet. Each node in the sequence-encoding layers, i.e., layers at height $l > 0$, encodes an instance of an activation map from the level below. Hence, it is possible to recover the sequential properties of the nodes below that eventually correspond to state-action-reward triplets at the bottom level.

On-line reproduction is performed as the robot selects on every timestep one node at the bottom level and executes its encoded action. The algorithm selects the optimal node to activate based on a set of high-level requirements that are set by the user. Three parameters are devised to evaluate the fitness of each node at the bottom level. These parameters consider how well a particular action, if executed, would fit to the past performance of the agent,to the current observations of the environment, and to the expected discounted future reward.

1) Historical Match: The *historical match* calculation starts from the top of the hierarchy and continues towards the bottom level. Intuitively, the historical match is scalar approximation that represents if a particular node at any level is activated next how well the short-term memory activations of that level will match the long-term memory weights encoded above. This is achieved by considering the pattern differences between the LTM of a node at layer l and the STM at layer $l-1$. Nodes whose weights resemble better the activation map on the layer below have higher historical match. The calculation of the historical match is a complex procedure [10] as the differences between STM and LTM are usually temporally misaligned. For example, Figure 3 illustrates the progression of an activation map and how it is represented on the node above.

The STM activations on any layer change dynamically during reproduction, hence, the historical match of nodes needs to be updated accordingly on every step. The correct calculation of the historical match is a key factor towards the hidden state identification under partial observability.

2) Input Match: The Euclidean distance between the robot's immediate observation, i.e., its current state, with the node weights at the bottom layer. The closer a node is to the current observation the higher the value of the input match. The input match is projected in the continuous range of $[0, 1]$ corresponding to the node with maximum and the minimum Euclidean distance from the current observation.

3) Discounted Future Reward: The problem of finding the optimal action on timestep t requires additionally to consider an expected future value on timestep $t' > t$ for taking that

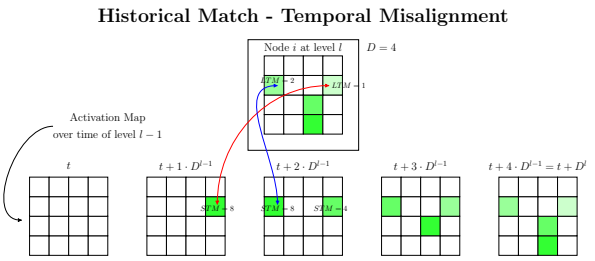


Fig. 3: A single node i on layer l encodes an activation pattern of the map on layer $l-1$. The encoded activation pattern corresponds to the future expected activation map on $t+D^l$ on the layer below, however, in the previous steps the activation pattern is temporally misaligned.

action. In the general case, the agents receive a delayed reward or insignificant reward before finally arriving at a goal state with a high reward. However, the algorithm needs to autonomously calculate the expected discounted future rewards. During the demonstrations, the agent receives from the environment a reward of 0.0 on every step and a positive scalar reward of 1.0 only on the final step. Setting the reward of the last activated node is trivial. However, nodes that were activated in the past steps are unaware of the final reward, due to the absence of direct lateral connections on any layer. Temporal information can be extracted though from the nodes on the layer above by exploiting the network of connections. Hence, it is possible to estimate the expected discounted future reward for any node. It first requires a bottom-up and then a top-down propagation of the reward to all nodes. At the end of the process, every connected node maintains the highest expected discounted future reward.

E. Action Selection

Each bottom level node is an action selection candidate. However, only the node with the highest *activation potential* is selected as a winner. The activation potential is a node-specific metric that accumulates the *historical match* and the *input match* in order to decide which node to activate next and execute its encoded action. We define the *activation potential* (AP) of a node at the bottom level as the weighted sum

$$AP = \lambda_{IM} IM + \psi_{HM} HM + \phi_{DR} DR, \tag{4}$$

with $\lambda_{IM} + \psi_{HM} + \phi_{DR} = 1.0$ to weigh the importance of each factor ($\lambda_{IM} \rightarrow$ input match (IM), $\psi_{HM} \rightarrow$ historical match (HM), and $\phi_{DR} \rightarrow$ discounted reward (DR)) depending on the desired properties of the agent. Depending on the application, the user sets a triplet of percentage weights for the action selection parameters. The bottom level node with the maximum activation potential is selected as the winner and the corresponding action is selected for execution. However, we have introduced a perturbation recovery method where the final executed action is a weighted sum of the dictated action at the bottom level and the expected state that robot should have been which is also encoded in the winning node.

F. Endogenous Generation of Rhythmic Movements

For the production of endogenous rhythmic movements, the contribution of the input match is minimized. Assuming that a fixed number of skills are already encoded in the hierarchy, a high historical match weight would correspond to being able to reproduce on demand any pattern by endogenously activating the node that would match the observed sequence. However, such a behavior is not sufficient for the development of complex behavior. Instead, a driving signal is required to exploit the use of the already encoded patterns; that signal is the *input match* which is not oscillatory, whereas a demonstrated pattern is a full period of an oscillatory movement.

The small contribution of the input match acts as a deciding factor in the beginning of the reproduction until the STM accumulates sufficient activations and the hierarchy can recall the remaining part of the primed pattern. Once completed, the STM activations are fully-decayed and the process begins again. Cycling through the possible patterns, we present initial findings on how complex behavior can emerge from selectively executing a movement from a repertoire of encoded skills.

G. Open Parameters

The Machine Learning community comes to an agreement that open parameters must be avoided whenever possible. Controlling parameters is not always intuitive and rarely easy for non-experts. HSOME relies on multiple variables manually configured by the user; however, the majority of them are systematically determined. The height of the hierarchy depends on the length of the demonstrations. The experiments in this work were all run with a fixed size of STM. Increasing the capacity of STM, D, influences the number of possible activation patterns to be produced at one layer and reduces the probability to reactivate the same node in the higher layers. This algorithm benefits from the wide literature on selecting appropriate SOM sizes, learning factors and neighborhood functions. The size of the bottom layer SOM relies on the observation space and the required descriptive resolution of the task. The number of nodes in the higher layers cannot be immediately predicted, however, it is possible to estimate a worst case scenario where no node reuse takes place. Being generous to ensure sufficient nodes are present in every layer to encode the complete sequences is the common practice in the presented experiments. The choice of the action selection parameters, i.e., selecting the weights λ_{IM}, ψ_{HM}, and ϕ_{DR} for the input match, historical match and expected discounted future reward respectively, is the most challenging for the user. Specific design decisions were made, e.g., scaling the IM, HM and DR values in the $[0, 1]$ range and setting the sum of these factors to be 1.0. As a result, the user selection reflects the percentage of influence for each parameter. Still, the results are sensitive to small changes in the weights. Exploring such trade-offs in the future is an interesting direction.

IV. Experimental Set-Up

A. Tactile Gesture - Tapping

Tactile gestures are motions that "involve brief, intentional contact to a relatively restricted location on the body surface of

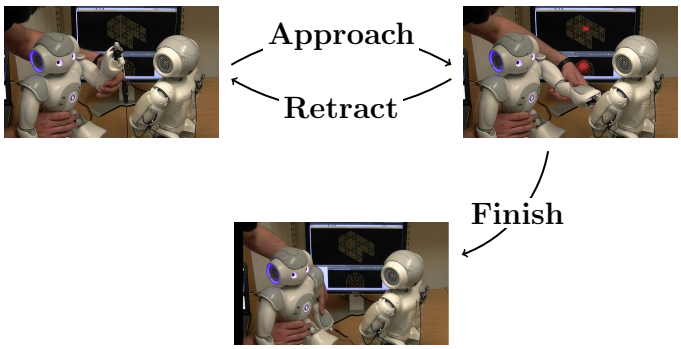


Fig. 4: A double tap is demonstrated to the robot. Starting from the top left configuration, the arm is extended to approach the other robot, retracted back, extended again, and then it is finally positioned to a resting configuration next to its left side.

the receiver during a social interaction" [45]. For humans the production of a tactile gesture may be trivial. On the contrary, robots need to be competent to produce such movements that involve multiple degrees of freedom and create tactile impressions that vary in time, location, area, and force. Regardless of the challenges faced, the rich expressiveness of tactile gestures in the interaction between co-located robots and humans is a strong motivation of this work.

In this work, we study the double tap skill from an active to a passive robot as it is a common tactile gesture between humans to draw attention of the other member. A successful double tap requires the *approach* and *retract* phases to be repeated two times. Hence, over the course of approaching and retracting, the algorithm observes similar states but on different directions. Hence, nodes with similar observations but opposing actions are encoded at the bottom level. A competent agent must carefully activate the corresponding node in order to successfully reproduce the double tap. However, the agent has partial observability and is not explicitly aware of the actions being executed as only the joint angle is known and not the joint velocity which implies the approaching or retracting phase. As a result, it is necessary for any algorithm to build a sufficient capacity of auto-associative memory in order to perform a sequence of actions that are influenced by the past actions as well. Another challenge the agent faces is the engagement to perform two taps but also be able to escape towards completing the task; a capability that is questionable if trapped in an attractor landscape. Finally, in a second experiment we demonstrate the effects of changing from historical match orientation to reward orientation. As it will be demonstrated, the agent reproduces a novel skill, i.e., a single tap, that has never been observed by the agent.

Figure 4 presents the experimental setup of the two tactile sensitive robots that are used to study robot to robot tactile gesture learning and reproduction. The gesture *producer* is equipped with a touch sensitive fingertip and the gesture *receiver* is equipped with large area tactile sensors, or robot skin. A human teacher is responsible for the kinesthetic demonstration of a double tap tactile gesture. The producer's touch sensitive fingertip approaches and retracts with its arm a tactile sensitive area in upper left arm of the gesture receiver. Kinesthetic teaching requires the teacher to manually guide the robot's arm while its motors are set to a passive mode. However, the joint values and contact feedback are recorded simultaneously. The robots maintain the same basic configuration throughout this experiment.

A *trial* consists of a pair of a single demonstration and one reproduction. Twenty trials were performed in total in each of the two experiments. The two experiments aim to study the reproduced gestures for faithful reproduction and reward oriented exploitation, namely the double tap and the single tap respectively. The same set of demonstrations is used for both experiments in order to train the hierarchy and then run a single reproduction. The sensor data of each demonstration were recorded at $\sim 50Hz$ and sub-sampled at $\sim 8Hz$ resulting in 102 samples for each demonstration.

A network architecture of height 5 with an STM capacity $D = 4$ is used to accommodate encoded sequences of up to 256 steps for both experiments. The bottom layer is a 12×12 node map and is trained for $100,000$ steps. The learning factor is linearly decreasing and the Moore neighborhood function (i.e., the two-dimensional square lattice surrounding a central SOM node) is also decreasing over time according to the learning factor. Concretely, the algorithm starts with $a(t_0) = 0.9$ with a Moore neighborhood function of range 3 (48-cell) until the learning factor reaches $a(t) = 0.5$. Thereafter, the Moore neighborhood decreases down to range 1 (8-cell) for $0.2 \leq a(t) < 0.5$ and finally the range becomes 0 (winner only) for $a(t) < 0.2$. The sequence-encoding layers, these are all layers for $l > 0$, are of size 8×8, 5×5, 4×4, and 4×4 nodes from the lower to the top. The sequence-encoding layers require only a single observation to encode an input pattern. Consequently, they use a learning rate $a = 1.0$.

1) Experiment A - Double Tap: The goal of this experiment is to demonstrate the hidden state identification mechanism in a complex robot interaction experiment. In the double tap gesture, the robot suffers from consecutive observations that are not descriptive enough of which actions to follow next. Hence, a focus on the high historical match in the action selection parameters will motivate the robot to remain faithful to the demonstrated gesture. Otherwise, the partially observable and noisy environment will distract the historical match building process. The action selection parameters were chosen to be $\lambda_{IM} = 0.1$, $\psi_{HM} = 0.9$, and $\phi_{DR} = 0.0$, where the indexes are used for convenience to the reader.

2) Experiment B - Single Tap: The goal of the second experiment is to introduce a different set of action selection parameters focused on reward orientedness. Even though the same training data is used, the agent will now try to exploit the demonstrated gesture in order to reach a goal faster. However, the goal is not just to reach the final state but also to achieve intentional contact with the passive robot. Hence, the robot must approach and tap the passive robot once before discovering an opportunity to move the arm towards the goal state. The action selection parameters were chosen to be $\lambda_{IM} = 0.45$, $\psi_{HM} = 0.495$, $\phi_{DR} = 0.055$. The historical match weight may be larger than the expected discounted

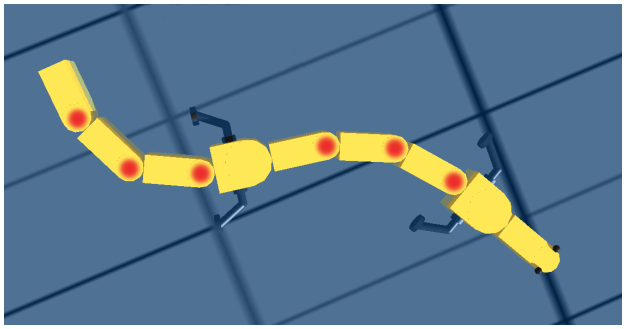


Fig. 5: The simulated salamander features 6 actuated hinges between the body parts. Between the head and the body part supporting the front limbs no actuator exists. The same applies for the node supporting the back limbs and the next body part towards the head. The original environment and the simulated model of the salamander is provided by [11].

future reward parameter; however, the absolute value of the historical match will drop drastically due to the mismatches. As a result, nodes with a good input match supported by higher expected discounted future rewards will motivate the agent to reach the goal state faster. The high contribution of the input match ensures that nodes with a high input match will be selected. At the same time, the movement will remain faithful at least to trajectories that have been observed in the demonstrations, regardless of any temporal inaccuracies.

B. Simulated Salamander

1) Environment and Robot: The publicly available simulated environment in Webots has been used in this experiment that features a swimming salamander-like robot [11]. Additionally, we have modified the environment with the addition of a permanent wall to separate the terrestrial from the aquatic environment forcing the robotic salamander to always remain within the limits of the aquarium. The salamander with its actuators highlighted in red is illustrated in Figure 5. Additionally, the salamander is equipped with two proximity sensors, one on each eye, that detects distances from objects in its frontal and side area.

2) Training Data: The original CPG controller [11], is used to record a long log of swimming demonstrations while avoiding obstacles at the same time. The difference of the proximity sensors coupled with the salamander's joint angles are recorded for sufficient time while swimming around the designated area. The goal is to record enough data that will cover a wide range of angles that the salamander encounters an obstacle, i.e., the surrounding wall, and avoids it. Concretely, on every timestep t the provided open-source oscillatory controller sets the angle θ_i of joint $i \in [0, 5]$ of the salamander:

$$\theta_{i,t} = A \sin\left(\phi_t + i\frac{\pi}{3}\right)\left(\frac{i+5}{10}\right) + \textit{spineOffset} \qquad (5)$$

where ϕ_t is the phase of the oscillation that is increased by a fixed amount in every simulation step. The first term of the sum produces an oscillatory movement with amplitude A and a phase difference of $\frac{\pi}{3}$ from one joint to the other. The phase difference is responsible for the wave-like propagation of movement across its joints. Furthermore, the amplitude of the motion of each joint i is proportional to its position in the chain, i.e., joints towards the tail have a larger amplitude; e.g., for joint $i = 5$ we have A and for the head $i = 0$ we have $A/2$. The *spineOffset* parameter is responsible for the turning of the salamander either to the right or left. The spineOffset also plays the role of the feedback signal, i.e., the difference between two obstacle detectors on the head of the salamander, that shapes the original oscillatory movement to develop more complex behavior. Hence, the salamander-like robot is able to avoid the walls of the aquarium by turning right or left while swimming.

Ten short oscillatory trajectories are extracted from the log along with the difference of the proximity sensors. The extraction is based on our experience that a full period of a movement from our recordings lasts 40 timesteps. The chunks of 40 steps were sorted based on the initial observation signal from minimum to maximum and ten chunks uniformly distributed in the range of the observation were selected. The observation is the difference of the reading of the right sensor minus the left sensor. Each sensor has a maximum reading of 0.7 when there is no obstacle and decreases down to 0.0 as it approaches an obstacle. Hence, the ten trajectories correspond to the discretized range of the difference of the proximity sensors, from the minimum to the maximum observed (min: obstacle only to the right, max: obstacle only to the left).

Each training sample is composed of the 2D input, i.e., the difference between the proximity sensors and the binary input of whether the salamander is swimming or walking (in this work the salamander is always in swimming mode). The action space is a 10D signal, i.e., 6DOF for the actuators between the salamander's body parts and 4DOF for the limbs; however, the limbs were also constant during this experiment. The learned actions are not joint differences to apply on every step but rather actual configurations that have to be applied to the robot.

3) Reproduction: After encoding the ten demonstrations, HSOME tries to develop swimming skills based only on the proximity sensor in the initial phase of each reproduction. The remaining steps of the cycle are reproduced without influence from the observed joint angles. This property of HSOME has not been demonstrated in the past, but reproducing patterns in the absence of input or observations is an attractive feature for learning algorithms [46]. In the first steps of the reproduction the input match is important, however, in the following steps the historical match suffices to drive the remaining reproduction. This is possible because the input match weight is $\lambda_{IM} = 0.01$, and the historical match weight is $\psi_{HM} = 0.99$. In the first step all trajectories may have an equal historical match value, however, the small contribution of the input match helps the selection of the best trajectory. Thereafter, the historical match internally motivates the corresponding sequences to be reproduced.

4) Configuration: A hierarchy of height 4 with an STM capacity $D = 4$ is used to accommodate encoded sequences of up to 64 steps for both experiments. The bottom level is a 5×5 node map, trained for $100,000$ steps. The learning factor

is linearly decreasing and the Moore neighborhood function is also decreasing over time according to the learning factor. Concretely, the algorithm starts with $a(t_0) = 0.9$ with a Moore neighborhood function of range 3 (48-cell) until the learning factor reaches $a(t) = 0.5$. Thereafter, the Moore neighborhood decreases down to range 1 (8-cell) for $0.2 \leq a(t) < 0.5$ and the final range becomes 0 (winner only) for $a(t) < 0.2$. The sequence-encoding layers, these are all the higher layers than the bottom, are of size 10×10, 7×7, and 5×5 nodes from the lower to the top. The sequence-encoding layers require only a single observation to encode an input pattern, therefore, they use a learning rate $a = 1.0$.

V. Experimental Results

A. Tactile Gesture

1) Results A - Double Tap: In Figure 6, the solid green and dashed blue lines represent the mean demonstrations and reproductions respectively. The surrounding envelope represents the 95% confidence interval expressed as $\bar{\xi} \pm 1.96\sigma$, where $\bar{\xi}$ is the mean demonstration (20 trials during kinesthetic teaching) or reproduction (20 trials) as observed in the joint angles through the encoders, and σ is the standard deviation.

On each trial, the robot starts from an initial configuration close to the initial one from each demonstration and then autonomously reproduces the double tap gesture. The characteristic of the double tap gesture is most obvious on the elbow roll of the robot (Figure 6(b)) that is mainly responsible for the repeating approach and retract of the arm. Additionally, the tactile impression is illustrated in the pressure profiles of both the fingertip (Figure 6(c)) and the skin (Figure 6(d)). The agent successfully reproduced the demonstrated trajectories. The action selection mechanism being influenced by the high historical matches supported HSOME to overcome the problem of immediate observations with null velocities at the contact point. Furthermore, the tap is repeated exactly two times before retracting the arm to its side, which also demonstrates that the agent is able to perform a transition from a repetitive motion to a discrete movement.

The skin pressure profiles (Figure 6(c) and 6(d)) show great similarity between the demonstrated motions and the reproduced ones. However, the pressure profiles during demonstration present lower values. We argue that multiple reasons exist for that disparity, however, the main reason is that the active robot is supported by the supervisor from its back during the demonstrations, while during reproduction the robot sits on its own legs with the hip joints playing supportive role. Kormushev et al. [47] also confirm our argument that kinesthetic teaching to free-standing robots interacting with other objects is challenging as the teacher produces external forces affecting the overall dynamics of the system.

The historical match is responsible for the hidden state identification process throughout the reproduction phase. An example of the historical match values of the winning nodes in a single trial is illustrated in Figure 7(a). The high contribution of the historical match during action selection is reflected on the progressive increase of the historical match. However, there are two periods, one being between steps 20 and 30, and the other between steps 40 and 65, that the historical match meets some resistance and does not grow. These two periods coincide with the two contact phases of the double tap and suggests that the increased noise contributes to the selection of other nodes that feature a better input match. Nevertheless, the successful selection of higher nodes does not allow the miss-matches on one layer to propagate above.

2) Results B - Single Tap: In Figure 8, the solid green and dashed blue lines represent the mean demonstrations and reproductions respectively. The surrounding envelope represents the 95% confidence interval expressed as $\bar{\xi} \pm 1.96\sigma$, where $\bar{\xi}$ is the mean demonstration (20 trials during kinesthetic teaching) or reproduction (5 most similar trials) as observed in the joint angles through the encoders, and σ is the standard deviation.

In this experiment, the robot starts from the same configuration with the double tap experiment and autonomously tries to reproduce the double tap gesture. However, now the decisions of the agent are reward oriented and remain less faithful to the initial demonstrations. As the arm approaches the contact point, the combination of input match, historical match and expected discounted future reward coincide on which actions are the best to be taken. At the contact point though, there are actually two types of actions that may be selected. Both types also have similar input matches but different historical match and expected discounted future reward. These are a) to retract the arm as part of initiating the second tap, or b) to retract the arm and move it towards the final configuration.

Finding a fine balance between the action selection parameters proved to be difficult for the humanoid robot control experiments, especially when trying to use a single set of weights over all trials of the experiments. Consequently, the algorithm was not successful in all 20 trials, nevertheless, it is important to present such limitations in our results.

In total, 20 trials were run for this experiment and approximately 8 succeeded to produce the single tap and arrive at the goal location. However, the 5 most similar reproductions in qualitative terms are presented, while the remaining 3 presented a long delay at the contact point; otherwise, given the small number of samples using all reproductions would skew the mean values. Of the remaining 12 reproductions, 7 trials did not exploit the reward and performed a complete double tap due to the low expected discounted future reward weight, and the other 5 failed completely to reproduce the motion as the agent moved the arm directly towards the final configuration without achieving a contact first, due to the bad selection of the parameters.

The difference between the single tap and the double tap gesture is better illustrated on the elbow roll joint (Figure 8(b)), the fingertip feedback (Figure 8(c)), and the skin sensor feedback (Figure 8(d)). Similarly, both the single and double tap skills are also illustrated in the task space in Figure 9(a). Figure 9(b) focuses on the contact point to highlight the difference between the demonstrated and the reproduced skill.

The novel reproduced tactile gesture is considerably different from the demonstrated one. Consequently, the behavior of the historical match accumulation must also differ. In Figure 7(b), the historical match of a single trial is presented. As expected, it does not increase progressively over time and

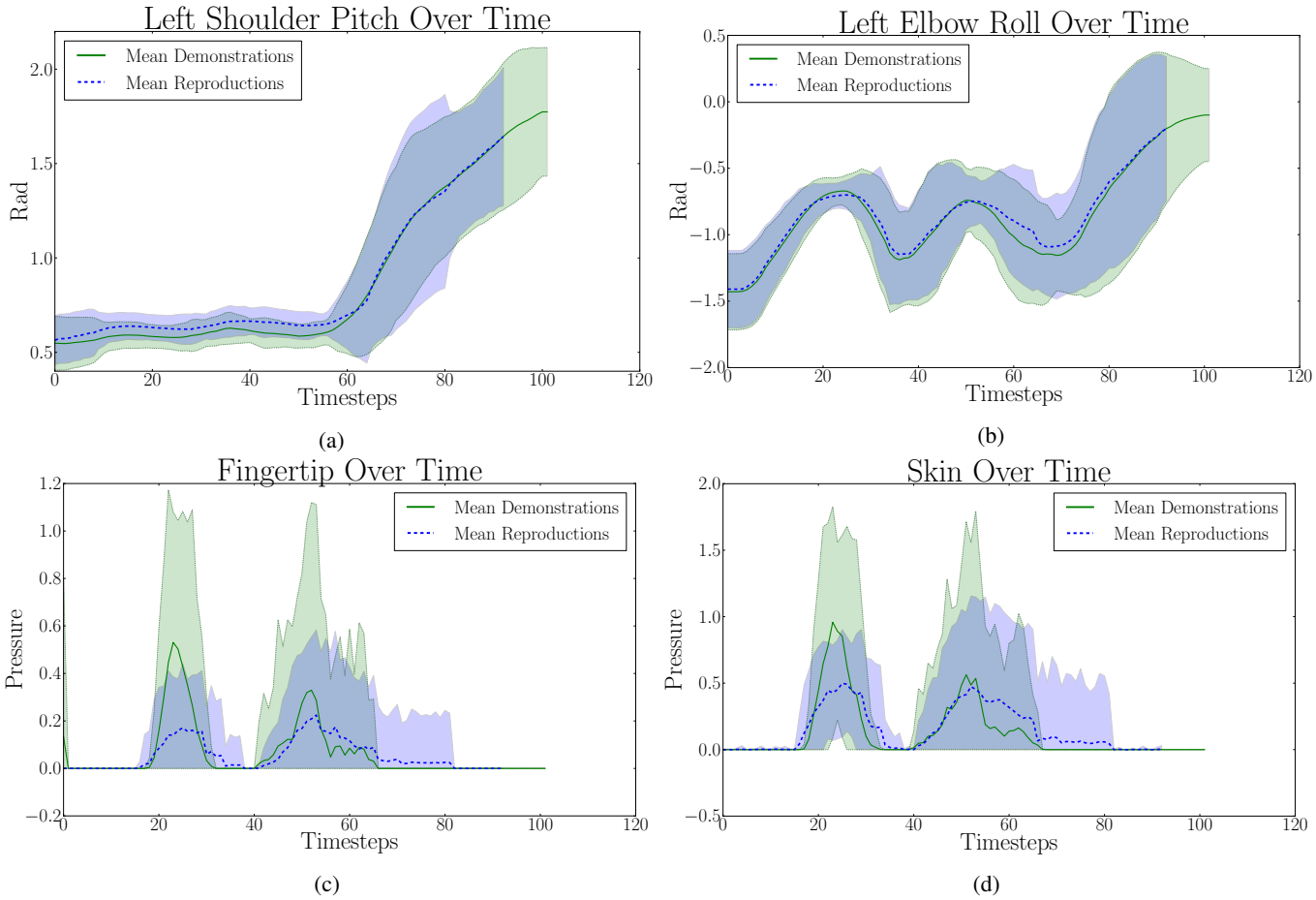
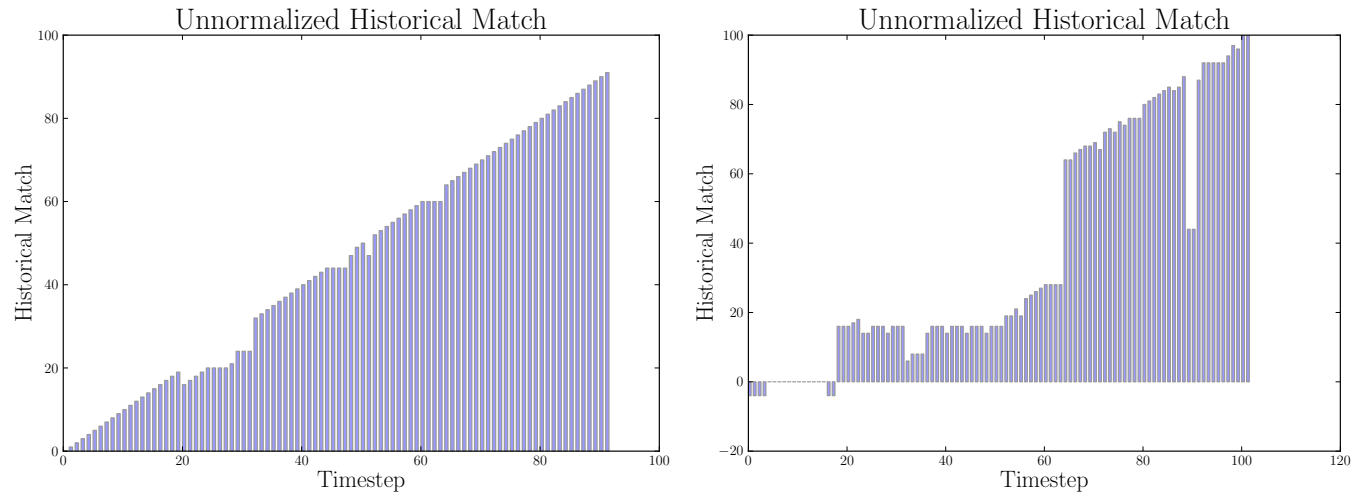


Fig. 6: The continuous green trajectory represents the demonstrations. In particular, the double tap pattern is most visible on the elbow roll joint in (b). The dashed blue line represents the reproduced trajectories. The surrounding envelopes represent the 95% confidence interval expressed as $\pm 1.96\sigma$. Taxel pressure values in (c) and (d) are not calibrated to physical pressure units. Sub-figures are temporally aligned in the vertical direction, hence, it is easy to compare where contact is achieved and lost (second row) throughout the range of the movement (first row).



(a) Double Tap: Historical Match Orientation

(b) Single Tap: Expected Discounted Future Reward Orientation

Fig. 7: Historical match progression of a single trial in (a) the double tap and (b) the single tap experiment. The historical match value on each time step corresponds to that of the winning node only. On every step the winning node is the one with the highest activation potential, i.e., the sum of input match, historical match, and expected discounted future rewards. Hence, the illustrated values do not necessarily represent the highest historical match of any node on every time step.

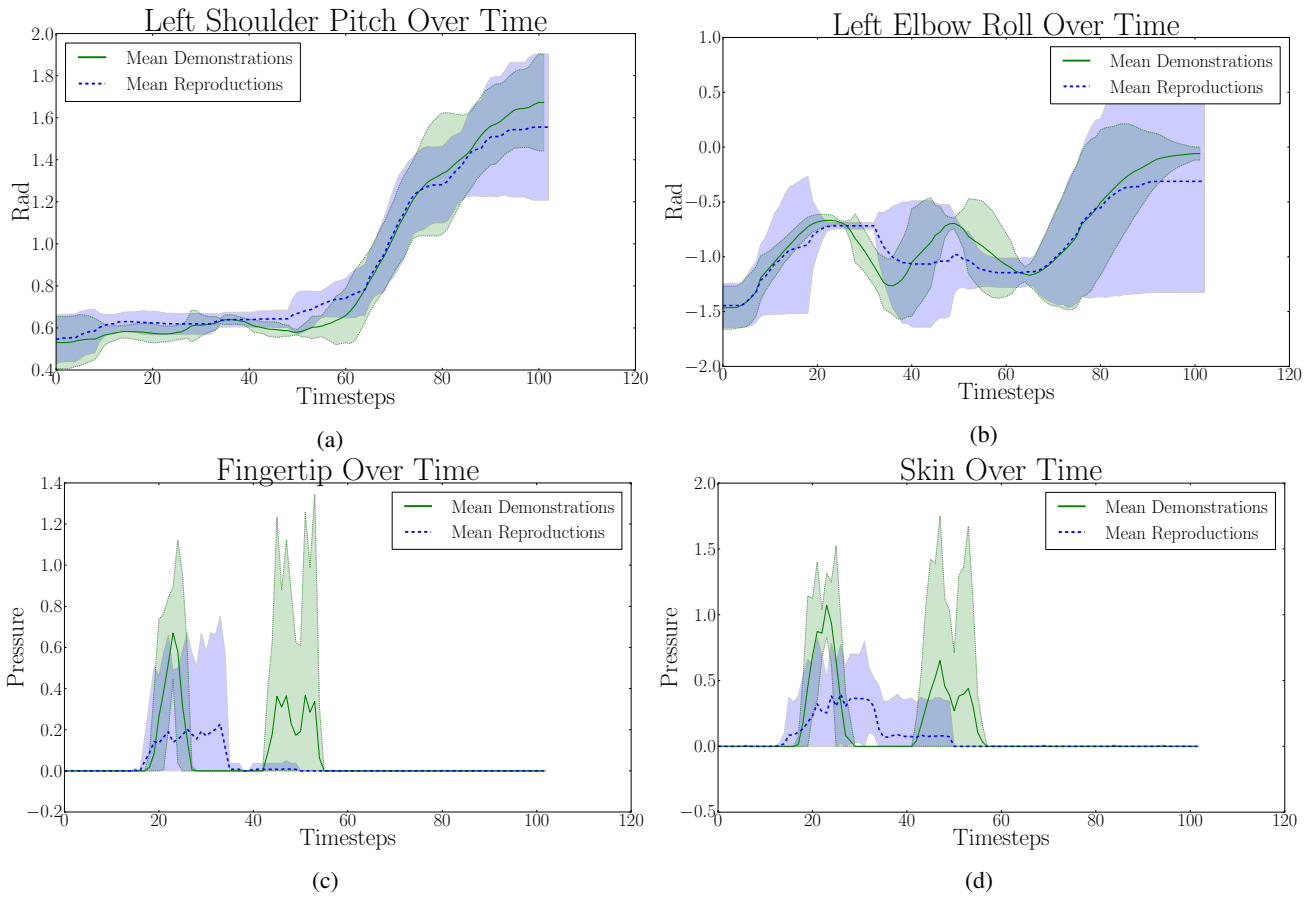


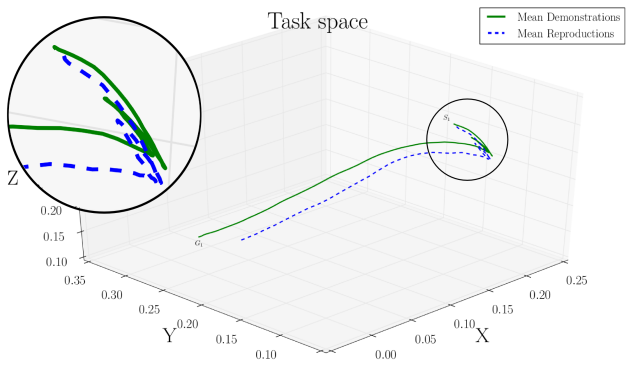
Fig. 8: The continuous green trajectory represents the mean demonstrations (over 20 trials) and the dashed line represents the mean reproductions (over 5 trials). The surrounding envelopes represent the 95% confidence interval expressed as $\pm 1.96\sigma$. In (c) and (d) it is better illustrated that the tactile impression is a single tap. Taxel pressure values in (c) and (d) are not calibrated to physical pressure units.

is worth comparing it with the double tap counterpart shown in Figure 7(a). As seen in Figure 7(b), the agent initially (steps 0 to 20) performs actions that do not build any HM but is rather motivated by the IM. After initiating contact (approximately after timestep 20, as verified by Figure 8(c)), the HM is gaining some value until timestep 32 as the behavior so far matches the demonstrated one. The algorithm would expect at timestep 32 to leave contact and perform another one right after, however, the agent maintains contact and a drop in HM is observed, which increases shortly after the previous levels, as if the agent is in the first contact. Even though the HM builds up again from timestep 37 to 50, the absolute value is almost half of what is observed during the same timesteps in the double tap in Figure 7(a). Hence, the agent may have built some HM, but by no means has the value of having following the expected trajectory. Thereafter, the HM is being built quickly and regardless of the low HM values in the beginning as towards the end the nodes that get activated in the higher layers dominate the HM values below.

3) Limitations: Tuning the parameters λ_{IM} and ψ_{HM} intuitively is not a difficult task and based on our experience the results are not sensitive to their selection. However, balancing the weight ϕ_{DR} with ψ_{HM} is challenging in non-abstract domains. Handling hidden states while exploiting actions that may yield a higher expected discounted future reward requires a fine balance that can only be found by trial and error. Even then the results are sensitive to small changes in the weights as observed in the single tap experiment, hence exploring in the future automatic ways to dynamically balance this trade-off is an essential direction.

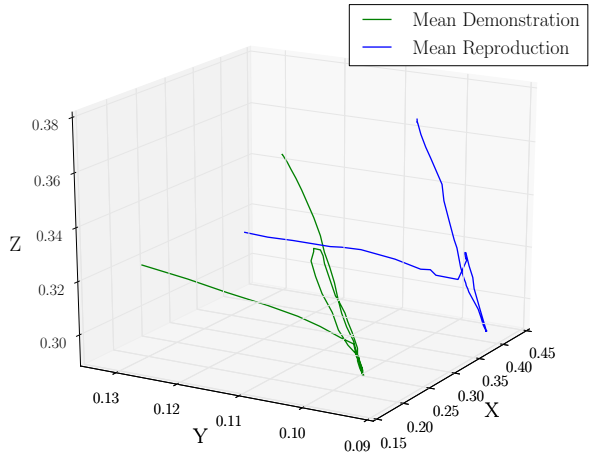
B. Simulated Salamander

In the reproduction phase, the trained hierarchy is responsible for controlling the simulated salamander repeatedly by selecting and endogenously reproducing each trajectory based on the initial observation. At the end of each reproduced trajectory, the STM is reset and the same process is repeated. Hence, for the length of a single period that each trajectory is being reproduced, the input signal does not influence the behavior. However, this may result in delayed reaction to obstacles until the input signal to the algorithm can play a decisive role again in the trajectory selection. Additionally, the environment is partially observable in the sense that only the difference of two proximity sensors is being used as input. As a result, when the salamander is stuck in corners the difference of the signals is the same as if it was swimming



(a)

Task Space: Double Tap Demonstration - Single Tap Reproduction



(b)

Fig. 9: In (a), the demonstration of the double tap tactile gesture along with its respective reproduction is presented in the 3D task space. The green continuous line illustrates the mean spatial location of the gesture producer's fingertip during the demonstrations. The dashed blue line corresponds to the mean reproduced trajectories in the task space during experiment. The uncertainty envelopes have been omitted in the 3D graph for better clarity. In (b), the region near the contact point of the double tap demonstration with the single tap reproduction is presented in the 3D task space. The green continuous line represents the mean spatial location of the gesture producer's fingertip as observed during the demonstrations. The blue trajectory corresponds to the reproduced trajectories (5 trials) in the task space. The reproduction trajectory is shifted by $+0.2$ on the x-axis for better clarity.

free. Another challenge is that as the salamander swims, it swings its head left and right giving "faulty" input signals in the sense that the angle of the body against the wall is derived by the salamander's head at the beginning of a period, and not say from the sagital plane of the body. Consequently, the swimming behavior that emerges from the salamander also presents obstacle-avoidance capabilities, however, given the aforementioned challenges of the environment it cannot be considered a fully developed behavior. These obstacle avoidance capabilities are illustrated in Figure 10 that presents the locations of the salamander in the aquarium as sensed from an on-board simulated GPS. Each sub-figure corresponds to the complete trajectory up to the reproduction time step stated on its title. The last portion of the swimming trajectory (2000 steps) that was generated is presented in red for clarity.

An example of the reproduced joint angles of the salamander are illustrated in Figure 11. As expected the joint angles are quantized. However, as seen from the observation O signal, the trajectories change according to the proximity sensor values to facilitate the obstacle-avoidance capabilities. In our experiments the original controller outperformed the learned agent but it is important here to show that the algorithm is capable of encoding and reproducing oscillatory skills in order to build more complex behavior such as swimming and avoiding walls. Ideally, the agents must further improve or even generalize the emerged behavior. In the current version of HSOME there are no mechanisms in place to allow such functionality. In the future we plan to devise a secondary logging system in order to reuse the observations and actions as training data for additional off-line training.

VI. Conclusions

Based on the CLA architecture [35], we have introduced HSOME [9], [10], a state-of-the-art algorithm in the area of Hierarchical SOMs for learning robot control. In this work, we have contributed with two additional experimental studies and a new reproduction configuration of HSOME to support endogenous generation of rhythmic movement patterns.

In summary, HSOME demonstrated the encoding and reproduction of a more complex tactile gesture, namely the double tap skill overcoming multiple hidden states along the reproduction trajectory. Additionally, HSOME developed a novel skill in a reward oriented reproduction that has not been observed before. However, the experiment was not fully successful. Physical humanoid robot control proved to be a difficult domain for the fine adjustment of the action selection parameters. However, the presented results motivate us to further study the novel skill acquisition by RL, in the direction of automatically identifying the optimal action selection parameters. On the other hand, learning oscillatory behaviors rely on building temporal associations between observations and actions. Hence, a traditional SOM would not have been able encode them. Instead, the simulated salamander experiment has successfully demonstrated the use of HSOME in learning oscillatory behaviors by demonstration. With this experiment, we present for the first time the encoding capabilities of HSOME to support ten different demonstrations in a single hierarchy. Additionally, the results presented the development of a wall-avoiding swimming capabilities by autonomously reproducing ten different swimming skills. The extensive covered area in the aquarium suggest that the emerged behavior is sufficient to avoid the surroundings of the aquarium. Nevertheless, the swimming speed of the salamander did not meet the speed of the original controller. We argue that the temporal order of execution of the various configurations is already optimal. Instead, an interesting direction to study in the future is the improvement of the encoded skills by adjusting

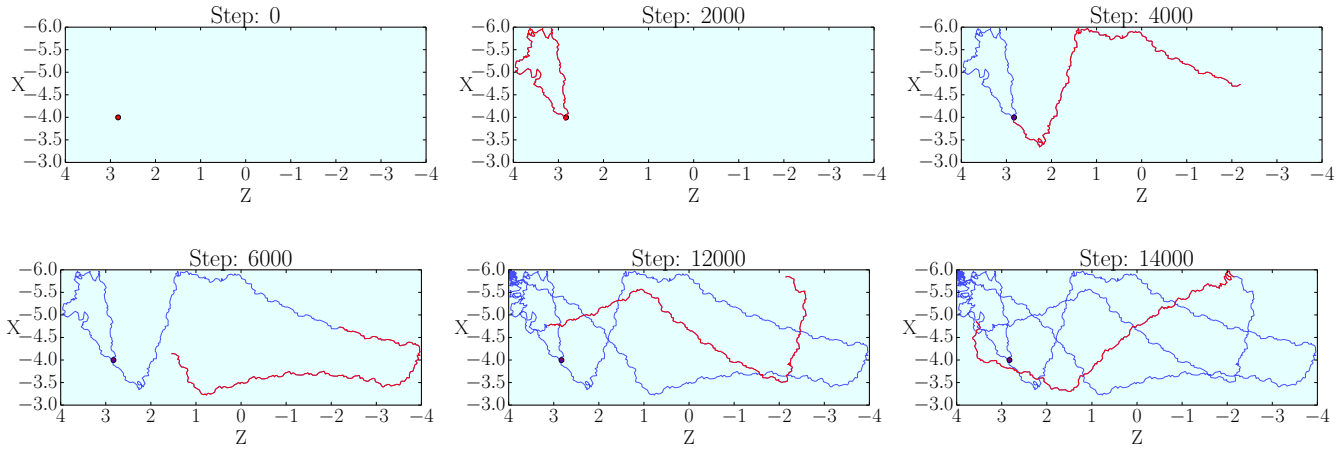


Fig. 10: The swimming trajectory of the simulated robotic salamander. The salamander starts at the red location shown at the top left graph. The red trajectory on each graph represents the locations of the last 2000 steps for clarity and the blue represents all past locations.

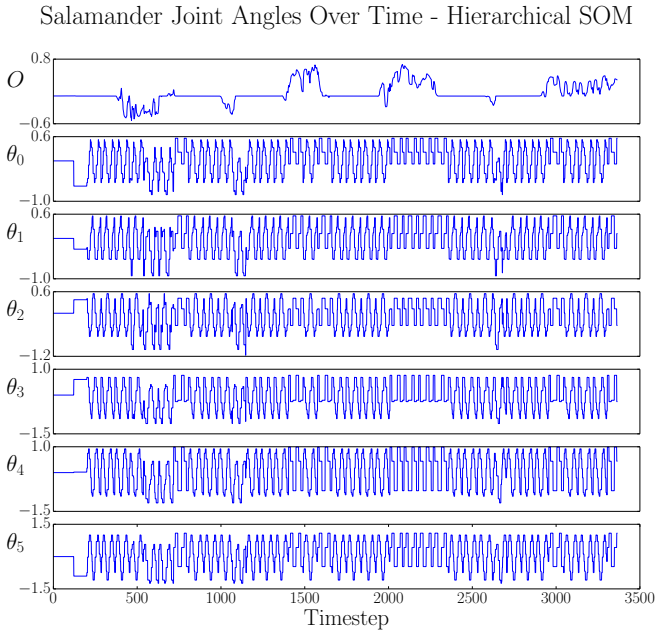


Fig. 11: In the first row, the observation signal is illustrated, i.e., the difference of the two proximity sensors positioned at the eyes of the salamander. The remaining 6 rows correspond to the angles over time from head to tail of the salamander. Depending on the observation signal, different oscillatory patterns are reproduced from the salamander.

the bottom layer weights through traditional RL. With a trivial reward function based on the swimming speed of each periodic movement and the well defined simulated environment; we hypothesize that an RL agent could significantly improve the swimming speed by autonomously crafting the configurations of the salamander while iterating from one bottom level node to the other. Hence, these adjustments correspond to the actual configurations of the salamander.

REFERENCES

[1] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Advances in neural information processing systems*, 2009, pp. 849–856.

[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.

[3] M. Schlesinger and B. McMurray, "The past, present, and future of computational models of cognitive development," *Cognitive Development*, vol. 27, no. 4, pp. 326–348, 2012.

[4] D. Caligiore, D. Parisi, and G. Baldassarre, "Integrating reinforcement learning, equilibrium points, and minimum variance to understand the development of reaching: A computational model." *Psychological review*, vol. 121, no. 3, p. 389, 2014.

[5] V. Meola, D. Caligiore, V. Sperati, L. Zollo, A. Ciancio, F. Taffoni, E. Guglielmelli, and G. Baldassarre, "Interplay of rhythmic and discrete manipulation movements during development: A policy-search reinforcement-learning robot model," *IEEE Transactions on Cognitive and Developmental Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[6] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[7] R. Yuste, J. N. MacLean, J. Smith, and A. Lansner, "The cortex as a central pattern generator," *Nature Reviews Neuroscience*, vol. 6, no. 6, pp. 477–483, 2005.

[8] L. B. Cohen, H. H. Chaput, and C. H. Cashon, "A constructivist model of infant cognition," *Cognitive Development*, vol. 17, no. 3-4, pp. 1323–1343, 2002.

[9] G. Pierris and T. Dahl, "Humanoid tactile gesture production using a hierarchical SOM-based encoding," *Autonomous Mental Development, IEEE Transactions on*, vol. 6, no. 2, pp. 153–167, June 2014.

[10] ——, "A developmental perspective on humanoid skill learning using a hierarchical SOM-based encoding," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 708–715.

[11] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[12] A. Crespi, K. Karakasiliotis, A. Guignard, and A. Ijspeert, "Salamandra robotica ii: An amphibious robot to study salamander-like swimming and walking gaits," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 308–320, April 2013.

[13] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, August 2007, pp. 702–707.

[14] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. Lausanne, Switzerland: EPFL Press, 2009.

[15] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[16] K. Doya, "Reinforcement learning in continuous time and space," *Neural computation*, vol. 12, no. 1, pp. 219–245, 2000.

[17] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, p. 0278364913495721, 2013.

[18] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1-2, pp. 171–203, 2011.

[19] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot iCub," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 417–423.

[20] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013. [Online]. Available: http://www.mdpi.com/2218-6581/2/3/122

[21] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[22] F. Stulp and O. Sigaud, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[23] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[24] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0921889008001772

[25] S. P. Chatzis and Y. Demiris, "Echo state gaussian process." *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1435–1445, 2011.

[26] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *ICML*. Omnipress, 2011, pp. 1017–1024.

[27] R. W. Paine and J. Tani, "Motor primitive and sequence self-organization in a hierarchical recurrent neural network." *Neural networks : the official journal of the International Neural Network Society*, vol. 17, no. 8-9, pp. 1291–309, 2004.

[28] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment," *PLoS Comput Biol*, vol. 4, no. 11, p. e1000220, 11 2008. [Online]. Available: http://dx.doi.org/10.1371\%2Fjournal.pcbi.1000220

[29] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[30] D. Prokhorov, "Echo state networks: appeal and challenges," in *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3, july-4 aug. 2005, pp. 1463 – 1466.

[31] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[32] G. J. Chappell and J. G. Taylor, "The temporal kohonen map," *Neural Netw.*, vol. 6, no. 3, pp. 441–445, Mar. 1993. [Online]. Available: http://dx.doi.org/10.1016/0893-6080(93)90011-K

[33] M. Varstal, J. Milln, and J. Heikkonen, "A recurrent self-organizing map for temporal sequence processing," in *Artificial Neural Networks ICANN'97*, ser. Lecture Notes in Computer Science, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Springer Berlin / Heidelberg, 1997, vol. 1327, pp. 421–426, 10.1007/BFb0020191. [Online]. Available: http://dx.doi.org/10.1007/BFb0020191

[34] T. Voegtlin, "Recursive self-organizing maps," *Neural Networks*, vol. 15, no. 89, pp. 979 – 991, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608002000722

[35] H. H. Chaput, "The constructivist learning architecture: A model of cognitive development for robust autonomous robots," Ph.D. dissertation, Department of Computer Sciences, The University of Texas at Austin, August 2004, also Technical Report TR-04-34. [Online]. Available: http://nn.cs.utexas.edu/?cla

[36] T. Ogata, K. Hayashi, I. Kitagishi, and S. Sugano, "Generation of behavior automaton on neural network," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2, 7-11 1997, pp. 608 –613 vol.2.

[37] A. Shimada and R. Taniguchi, "Gesture recognition using sparse code of hierarchical SOM," in *Pattern Recognition (ICPR), International Conference*, 2008, pp. 1–4.

[38] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biology*, vol. 11, no. 23, pp. R986 – R996, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960982201005814

[39] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[40] I. Delvolvé, P. Branchereau, R. Dubuc, and J.-M. Cabelguen, "Fictive rhythmic motor patterns induced by nmda in an in vitro brain stem–spinal cord preparation from an adult urodele," *Journal of Neurophysiology*, vol. 82, no. 2, pp. 1074–1077, 1999.

[41] A. Roberts, S. R. Soffe, E. S. Wolf, M. Yoshida, and F.-Y. Zhao, "Central circuits controlling locomotion in young frog tadpoles," *Annals of the New York Academy of Sciences*, vol. 860, no. 1, pp. 19–34, 1998.

[42] G. Guimarães, V. S. Lobo, and F. Moura-Pires, "A taxonomy of self-organizing maps for temporal sequence processing," *Intelligent Data Analysis*, vol. 7, no. 4, pp. 269–290, 2003.

[43] S. Marsland, *Machine learning: an algorithmic perspective*. CRC Press, 2011.

[44] K. N. Gurney, *An introduction to neural networks*. CRC Press, 1997.

[45] I. Morrison, L. S. Löken, and H. Olausson, "The skin as a social organ." *Experimental Brain Research*, vol. 204, no. 3, pp. 305–314, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19771420

[46] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks*, vol. 16, no. 1, pp. 11–23, 2003.

[47] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3970–3975.

Georgios Pierris received the Diploma in Electronic and Computer Engineering (ECE) from the Technical University of Crete (TUC), Greece in 2009. He received a PhD from the University of Wales in 2015. Dr Pierris was introduced in the research area of robotics in 2007 and has been working in human-robot interfaces and humanoid motion to develop complex soccer skills for RoboCup competitions. His interest later shifted to cognitive robotics and, under the ROBOSKIN project, Dr Pierris developed novel biologically inspired algorithms for tactile gesture learning and reflexes. Dr Pierris is currently a Research Associate at the National Centre for Scientific Research, "Demokritos" in Greece developing scalable software platforms for "Internet of Things" Applications. Dr Pierris has been an Associate Fellow of RobotDoc, the Marie Curie doctoral training network in developmental robotics, and a member of the EUCog network, a European network for the advancement of artificial cognitive systems and robotics.

Torbjørn S. Dahl is a Lecturer (Assistant Professor) at Plymouth University, UK and a member of the Centre for Robotics and Neural Systems. He received an MEng in Artificial Intelligence and Knowledge Engineering from Imperial College, London in 1997 and a PhD from the University of Bristol in 2002. Dr Dahl has worked as a researcher at Hewlett-Packard Labs, Bristol, at the University of Southern California and at the Norwegian Defence Research Establishment (FFI). From 2004 to 2012 Dr Dahl was a Reader and the Founding Director of the Cognitive Robotics Research Centre at the University of Wales, Newport. Dr Dahl has been a principal investigator on both national (EPSRC) and European (FP7) research projects as well as Lead Academic on a knowledge transfer project with Sony's Technology Centre, UK. Dr Dahl is a member of the IEEE and ACM.